

Outside this region is the part of image which is noise free. The multiplied result is known as masked repair subimage. If a soft noise mask was requested then the soft region of the mask will reduce the value of the original pixels underneath by a multiplicative factor which has a value from near 0 near the noise edge to near 1 a few pixels away from the edge of the noise mask and has values between 0 and 1 in between. The multiplied result is known as masked repair subimage.

In further detail, when multiplying a repair subimage and a soft edged noise mask, the pixel value of each of the pixels of the repair subimage is multiplied with a corresponding value for each pixel of the soft edged noise mask according to the following equation:

$$m \times r_0,$$

where m is the value of a particular pixel of the soft noise mask, and r₀ is the pixel value of each of the corresponding pixels of the repair image.

The following steps 6A through 13A represent the core part of the method. These steps will be performed that number of times already selected by the user when the sample and repair subimages were being selected.

At step 6A, the masked repair subimage is split into two images, one which is high pass filtered (HR) and one which is low pass filtered (LR). Any low pass filter can be used for this step. Only HR is passed onto the next step, step 7A. The low pass filtered masked repair subimage is not used in step 7A, but instead is used in step 12A.

At step 7A the fast fourier transform of the high pass filtered masked repair subimage HR is calculated to generate magnitude M₂ and phase P₂. Magnitude M₂ and phase P₂ are provided to the next step, step 8A.

At step 8A, the minimum of magnitudes M₁ and M₂ is calculated. M₁ is the magnitude of the fast fourier transform of the sample subimage and M₂ is the magnitude of the fast fourier transform of the masked repair subimage. The minimum of magnitudes M₁ and M₂ is taken as a new value of the fourier transform, and such value together with (unmodified) phase information P₂ is provided to the next step of the method, step 9A.

At step 9A, the inverse fast fourier transform is calculated using the magnitude and phase output step 8A.

At step 10A, the calculated inverse fast fourier transform values are made real, that is, any imaginary component in such values are simply dropped. Since the input to step 10A is a matrix of numbers, some of which may be complex numbers, for such entries in the matrix, the imaginary components are set to 0. In addition, since the real values may be outside the range of 0 to 255, if an entry is less than 0, the entry is reset to be 0. If the entry is more than 255, the entry is reset to be 255. Otherwise, the real value is left untouched.

At step 11A, the pixels output by step 10A are mixed with the pixels of the high pass filtered masked repair subimage created in step 6A. The values that are far away from the mask should be reset to the values from the masked repair subimage because they are known to be noise free. The values that are in the region of the mask that is 0 should be left as is because the original masked repair subimage is known to be noisy there and we have just calculated better values for this region. For values of the soft masks that are between 0 and 1 there will be a mixing of values just calculated in previous step and those in the masked repair subimage. All these three types of operations (for soft mask value 0, 1 and between 0 and 1) can be defined in the following way. If mask value is m at a point, then the new pixel at that point will be : $m \times rp + (1-m) \times r$ where rp is the

value at that point in the high pass filtered masked repair subimage R and r is the value at that point in the output of previous step 10A.

In further detail, when combining a new high pass filtered repair subimage with a high pass filtered repair subimage using a soft edged noise mask or a binary noise mask, the pixel value of each of the pixels of the new high pass filtered repair image is combined with the pixel value of each of the corresponding pixels of the high pass filtered repair subimage using the value of the soft noise mask or the binary noise mask, for each pixel, according to the following equation:

$$m \times rp + (1-m) \times r,$$

where m is the value of the soft edged noise mask,

rp is the pixel value of each of the pixels of the high pass filtered repair subimage (HR), and

r is the pixel value of each of the pixels of the new high pass filtered repair subimage.

At step 12A, the output of the previous step 11A and the low pass filtered masked repair subimage LR from step 6 are merged. Merging consists of simply adding these two images. In further detail, when merging the high pass filtered repair subimage (obtained by combining a new high pass filtered repair subimage with a high pass filtered repair subimage using a soft edged noise mask or a binary noise mask) with the low pass filtered repair subimage, the high pass filtered repair subimage is simply added to the low pass filtered repair subimage. The output of this merging is provided to the next step, step 13A.

At step 13A, if the value of any pixel output by step 12A is greater than 255, the value is reset to 255. Otherwise leave it untouched. This completes the core part of the method. As described about, this core part is to be performed a set number of times, since the user has already selected the number of times this is to be performed at the beginning, when the repair and sample subimages were being selected.

At step 14A, it is determined whether the required number of iterations of steps 6A–13A have been performed. If not, the process is returned to step 6A for another iteration. If so, then the process continues to step 15A as shown in FIG. 11.

At step 15A, the user determines whether the user is satisfied with the results as displayed on the display device 10. If so, the process is done. If not, the user returns to step 1A to repaint the noise mask over those areas that are visually unacceptable. In such a case, new repair and sample subimages may have to be selected.

To summarize the two new methods and apparatus for practicing the methods described here, we have described fast iterative algorithms for image noise removal. While most existing algorithms have worked solely in spatial or frequency domain, our algorithms work in both domains, making it possible to fully exploit the advantages from each domain. Although a few previous algorithms combine frequency and spatial domain information, they required the image to be band limited and required that the band limits be known. Our algorithms do not have this limitation.

Our dual-domain approach can (1) reconstruct many contiguous noisy pixels, (2) reconstruct textures even when they are large featured, (3) maintain sharpness, (4) maintain continuity of features (e.g., lines) across the noisy region. In addition the algorithm for removing noise from non-uniformly shaded images allows noise removal from images which contain objects that have slowly varying intensity changes in addition to the textures and prominent lines. These advantages make the methods very useful in many areas.